

【特許請求の範囲】

【請求項 1】 命令フィールドと、処理対象のレジスタ名を記載した第 1 のフィールドと、処理対象の他のレジスタ名又は即値を記載した第 2 のフィールドで構成される命令コードを有する命令セットを解釈して実行するプロセッサの命令制御機構であって、
前記命令フィールドの値を解釈する命令デコーダと、
前記処理対象のレジスタ名及び前記処理対象の他のレジスタ名で指示されるレジスタと、
前記即値を記憶する記憶回路と、
前記レジスタ又は前記レジスタと前記記憶回路に記憶された値に対して、前記命令デコーダの解釈結果に対応する処理を行う演算回路とを備えるプロセッサの命令制御機構において、
前記第 2 のフィールドに前記処理対象の他のレジスタを記載したレジスタ命令コードは、前記命令フィールドを特定の値とし、前記第 2 のフィールドのうちの前記処理対象の他のレジスタを記載した以外の部分に第 2 の命令フィールドを記載することにより表され、
前記第 2 のフィールドに前記即値を記載した即値命令コードは、前記命令フィールドに前記特定の値以外の値が記載され、
前記即値と前記処理対象の他のレジスタを使用する以外は同一の処理である少なくとも一部の即値命令コードと即値類似レジスタ命令コードには、前記命令フィールドと前記第 2 の命令フィールドに同一の値が割り当てられ、
前記命令フィールドが前記特定の値であるかを判定するレジスタ命令検出回路と、
該レジスタ命令検出回路の判定結果に従って、前記命令フィールドが前記特定の値である時には前記第 2 の命令フィールドが、前記命令フィールドが前記特定の値でない時には前記命令フィールドが、前記命令デコーダに入力されるように切り換える選択回路を備えることを特徴とするプロセッサの命令制御機構。
【請求項 2】 請求項 1 に記載のプロセッサの命令制御機構であって、
前記即値命令コード及び前記即値類似レジスタ命令コードと類似した処理であるが、前記即値及び処理対象の他のレジスタの値は使用しない第 2 のレジスタ命令コードの少なくとも一部には、前記命令フィールドが前記特定の値以外の第 2 の特定の値で、前記第 2 の命令フィールドに前記即値類似レジスタ命令コードの前記第 2 の命令フィールドの値と同一の値が割り当てられ、
前記レジスタ命令検出回路は、前記命令フィールドが前記第 2 の特定の値であるかも判定するプロセッサの命令制御機構。
【請求項 3】 請求項 1 に記載のプロセッサの命令制御機構であって、
前記命令セットは、前記即値命令コードと前記即値類似

レジスタ命令コードの少なくとも一部に対応し、前記即値命令コードと即値類似レジスタ命令コードの基本長命令コードより短い短縮長命令コードを有し、
該短縮長命令コードの命令フィールドは、前記基本長命令コードの命令フィールドと同じ位置で、処理内容が類似の前記即値類似レジスタの前記第 2 の命令フィールドと同一の値を有し、
当該命令制御機構は、前記基本長命令コードを処理する基本長モードと前記短縮長命令コードを処理する短縮長モードを切り換えるモード切り換え回路を備え、
前記命令デコーダは、モードに応じて解釈処理を変化させるプロセッサの命令制御機構。
【請求項 4】 請求項 2 に記載のプロセッサの命令制御機構であって、
前記命令セットは、前記即値命令コードと前記即値類似レジスタ命令コードの少なくとも一部に対応し、前記即値命令コードと即値類似レジスタ命令コードの基本長命令コードより短い短縮長命令コードを有し、
該短縮長命令コードの命令フィールドは、前記基本長命令コードの命令フィールドと同じ位置で、処理内容が類似の前記即値類似レジスタの前記第 2 の命令フィールドと同一の値を有し、
当該命令制御機構は、前記基本長命令コードを処理する基本長モードと前記短縮長命令コードを処理する短縮長モードを切り換えるモード切り換え回路を備え、
前記命令デコーダは、モードに応じて解釈処理を変化させるプロセッサの命令制御機構。
【請求項 5】 請求項 3 に記載のプロセッサの命令制御機構であって、
前記短縮長モード時に、レジスタ命令検出回路が前記命令フィールドが前記特定の値であることを検出した時には、前記命令デコーダは、当該命令コードを前記基本長命令コードとして処理するプロセッサの命令制御機構。
【請求項 6】 請求項 4 に記載のプロセッサの命令制御機構であって、
前記短縮長モード時に、レジスタ命令検出回路が前記命令フィールドが前記特定の値又は前記第 2 の特定の値であることを検出した時には、前記命令デコーダは、当該命令コードを前記基本長命令コードとして処理するプロセッサの命令制御機構。
【請求項 7】 請求項 5 又は 6 に記載のプロセッサの命令制御機構であって、
レジスタ間転送命令を、前記第 2 のレジスタ命令コードに割り当てたプロセッサの命令制御機構。
【発明の詳細な説明】
【0001】
【発明の属する技術分野】本発明は、効率的な命令セットを簡単な機構で実行できるプロセッサの命令制御機構に関し、特に命令コードを効率的に圧縮し、プログラムサイズを小さくできる短縮長命令が使用できると共に多

数の命令を効率よく使用できる命令セットを実行できるプロセッサの命令制御機構を簡単な機構で実現する技術に関する。

【 0 0 0 2 】

【従来の技術】プロセッサでは実行できる命令セットが決められている。命令セットを構成する命令コードは、命令フィールド（オペレーションコード）とオペランドから構成される。命令フィールドは、処理の内容を示す部分で、オペランドは処理の対象となるレジスタやメモリアドレスを示す部分で、処理に使用する数値であることもある。このような数値は即値又はリテラルと呼ばれる。

【 0 0 0 3 】命令には、ジャンプ命令などのようにオペランドが 1 個の命令や、2 個のオペランドの値に対して加算などの演算を行ってその結果を一方のオペランドに格納したり、2 個のオペランドの一方から他方に値を移動させるなどのオペランドが 2 個の命令や、2 個のオペランドの値に対して加算などの演算を行ってその結果を別のオペランドに格納するなどのオペランドが 3 個の命令がある。なお、この 3 個のオペランドを使用する命令は、2 個のオペランドを使用する命令を組み合わせることにより実現できるので、3 個のオペランドを使用する命令を持たない命令セットもある。ここでは、上記の、オペランドを 1 個、2 個及び 3 個使用する 3 種のアドレッシングモードを有する命令セットを例として、以下の説明を行う。なお、レジスタのみを使用したアドレッシングモードの命令はレジスタ命令、即値を使用したアドレッシングモードの命令はイミディエート命令と呼ばれる。

【 0 0 0 4 】図 1 は、レジスタ命令と即値（イミディエート）命令の命令コードの従来の構成例を示す図であり、（1）がレジスタ命令の構成を、（2）が即値命令の構成を示す。図示のように、レジスタ命令と即値命令は同じ長さであり、本発明はこのようなレジスタ命令と即値命令の長さが同じ命令セットを対象とする発明である。上記のメモリアドレスを直接指示するメモリアドレス命令は、レジスタ命令や即値命令と長さが同じ場合も異なる場合もあるが、本発明はこれには直接関係しないので、いずれであってもよい。

【 0 0 0 5 】図 1 に示すように、レジスタ命令は、上位 6 ビットが命令フィールド OP 1 であり、次の 5 ビットが演算対象の第 1 のレジスタ R s 1 を示し、次の 5 ビットが演算結果を格納する結果格納先レジスタ R d を示す。ここまでは即値命令も同じである。レジスタ命令では、更に次の 5 ビットが演算対象の第 2 のレジスタ R s 2 を示すが、即値命令では次の 1 6 ビットが即値を示す。レジスタ命令の第 2 のレジスタ R s 2 の後の残りの 1 1 ビットは第 2 の命令フィールド OP 2 であり、OP 1 と合せて命令フィールドを形成する。従ってレジスタ命令は即値命令より種類が多い。

【 0 0 0 6 】図 2 は、図 1 のような命令セットを実行するプロセッサの命令制御機構の従来例の構成を示す図である。プログラムはメモリに格納されており、メモリから読み出された命令コードは、命令保持用フリップフロップ 1 1 から 1 4 に一時的に保持される。参照番号 1 1 が命令フィールド OP 1 の示す値を保持する部分で、1 2 が第 1 の演算対象レジスタ R s 1 を指示する値を保持する部分で、1 3 が第 1 の結果格納先レジスタ R d を指示する値を保持する部分で、1 4 がレジスタ命令の時には第 2 の演算対象レジスタ R s 2 を指示する値と第 2 の命令フィールド OP 2 を示す値を保持し、即値命令の時には即値を保持する部分である。フリップフロップ 1 1 と 1 4 に保持された命令フィールド OP 1 と OP 2 は命令デコーダ 1 5 に出力され、そこで解読されて、処理する内容が演算器 2 8 と 2 9 に指示されると共に、命令に応じて第 2 の演算対象レジスタ R s 2 の値を使用するか即値が使用されるかが選択される。フリップフロップ 1 2、1 3 及び 1 4（R s 2 の部分のみ）に保持された値により、レジスタ 1 8、1 9、1 7 が指示される。また、フリップフロップ 1 4 に保持された値は更にフリップフロップ 2 3 に保持される。命令デコーダ 1 5 の解読結果に応じて、レジスタ 1 9 の値又はフリップフロップ 2 3 に保持された値が内部バスに出力され、同時に出力されるレジスタ 1 8 と共に演算器 2 8、2 9 に入力され、そこで命令フィールド OP で指示された演算が行われる。その演算結果は、レジスタ 1 7 に格納される。このようにして 1 つの命令コードに対する処理が完了し、プログラムで指示される命令コードを順次行う。

【 0 0 0 7 】

【発明が解決しようとする課題】図 1 及び図 2 の例は、もっとも一般的な命令セット及びプロセッサの命令制御機構の従来例であるが、近年プロセッサの高速化・低価格化により、従来は専用回路を用いて行っていた様々な処理を、1 つのプロセッサでプログラムを切り換えながら処理できるようになってきた。その反面、プログラムサイズが膨大になり、キャッシュなどのチップ内蔵メモリや外部メモリが増大したり、それらのメモリのアクセス頻度が高くなるために消費電力が増大するといった問題が生じている。

【 0 0 0 8 】最近のように、メディア機器の携帯化が進むと、プロセッサのコストや消費電力は、機器トータルのコストや消費電力の増大によって大きな割合を占めるようになってきており、このため、いかに全体のプログラムサイズを圧縮するかがプロセッサの課題となっている。そのため、同じ長さの命令セットであれば、できるだけ多数の命令コードが表現できることが望ましい。

【 0 0 0 9 】また、このような課題を解決するための別の方法に、短縮長命令を使用することがある。短縮長命令を使用できれば、プログラムサイズを圧縮することが可能である。従来から基本長命令と共に短縮長命令が使

用できるようにサポートしているプロセッサがある。図 3 は、基本長命令と短縮長命令が両方使用できるようにした従来例における命令コードの従来例の構成を示す図である。この方法は、あらかじめ命令フィールド中に命令長を示すビットを持たせることにより、プログラム中の命令が基本長命令と短縮長命令のいずれであるかが識別できるようにする方法であり、図 3 では最上位ビットをこれに割り当てている。しかし、この場合、1 ビットがこのような識別に使用されることになり、その分他のフィールドのビットが減少することになる。図 3 では、命令フィールドの 1 ビットを基本長命令と短縮長命令の識別に割り当てており、命令フィールドの長さが 6 ビットから 5 ビットに減少している。6 ビットであれば 6 4 通りの命令が表されるのに、5 ビットであれば 3 2 通りの命令しか表せなくなる。基本長命令と短縮長命令の識別のための 1 ビットを他の部分、例えば、即値の部分の 1 ビットを基本長命令と短縮長命令の識別に割り当てることも考えられるが、その場合には即値の範囲が半分減少することになる。短縮長命令は、そもそも命令コードの長さが限られるため特に問題である。

【0010】そこで、プロセッサにモード切り換え機構を設け、プログラム中のモード切り換え命令でモードを切り換えられるようにする。短縮長命令モードであれば、命令コードはすべて短縮長命令であるとして処理される。図 4 は、モード切り換えを可能にした場合の命令コードの従来例の構成を示す図である。しかし、この方法では、当然のことながら短縮長命令に割り当てられなかった命令は使用できないが、すべての基本長命令セットを短縮長命令セットに割り当ててことは原理的に不可能であるため、基本長命令セットにはあるが短縮長命令としては使用できない命令が存在することになる。そのため、短縮長命令モードにおいてはこのような命令が使用できず、そのような命令を使用する場合には、一旦基本長命令モードに切り換える必要が生じる。これでは、プログラムが複雑で、サイズも大きくなる。

【0011】上記の 2 つの方法では、基本長命令も短縮長命令も命令フィールドは同じ長さであるとしたが、基本長命令セットと短縮長命令セットで命令フィールドの体系を大きく異ならせることにより、多くの命令を短縮長命令として持つことができるようになる。しかし、この場合、モード切り換えにより両方の命令セットを実行できるプロセッサでは、命令デコーダを共通に使用するためには、短縮長命令フィールドを基本長命令フィールドに変換するために、図 5 に示すような回路が必要になる。この回路は、短縮長命令モード時には、短縮長命令の命令フィールドを、対応する処理内容の基本長命令コードの命令フィールドに変換する命令コード変換エンコーダ 3 1 と、その出力を一時保持するフリップフロップ 3 2 で構成される。通常の処理においては、この変換回路での処理のためにパイプラインの 1 段分を消費する

か、あるいは、1 段分を消費しない場合でもデコーダのクリティカルパスを圧迫することになり、ハードウェアの増加を招くだけでなく、分岐ペナルティの増大で性能の低下を招くといった問題がある。

【0012】本発明は、コードの長さが同じ命令セットであれば、できるだけ多数の種類の命令コードが表現できるようにすることで、プログラムの長さを短くした命令コードのセットを実行できるプロセッサの命令制御機構を簡単な構成で実現することを目的とする。また、所定の長さのコードで多数の短縮長命令を効率よく実行できるプロセッサの命令制御機構を簡単な構成で実現することを目的とする。

【0013】

【課題を解決するための手段】上記目的を実現するため、本発明のプロセッサの命令制御機構は、命令コードのうち、処理対象のレジスタ名又は即値を記載した第 2 のフィールドにおいてレジスタ名を記載した場合には、第 2 のフィールドに使用しない第 2 の命令フィールドがあることに着目した。そして、第 2 のフィールドに処理対象のレジスタ名を記憶するレジスタ命令については、命令フィールドに特定の値を割り当て、実際の命令内容は第 2 の命令フィールドに記載するようにし、即値命令については、命令フィールドにこの特定の値以外の値を割り当て、しかも、即値とレジスタを使用する以外は同一の処理である少なくとも一部の即値命令コードと即値類似レジスタ命令コードには、命令フィールドと第 2 の命令フィールドに同一の値が割り当てて。これであれば、命令デコーダなどを複雑にすることもない。

【0014】すなわち、本発明のプロセッサの命令制御機構は、命令フィールドと、処理対象のレジスタ名を記載した第 1 のフィールドと、処理対象の他のレジスタ名又は即値を記載した第 2 のフィールドで構成される命令コードを有する命令セットを解読して実行するプロセッサの命令制御機構であって、命令フィールドの値を解読する命令デコーダと、処理対象のレジスタ名及び処理対象の他のレジスタ名で指示されるレジスタと、即値を記憶する記憶回路と、レジスタ又はレジスタと記憶回路に記憶された値に対して、命令デコーダの解読結果に対応する処理を行う演算回路とを備えるプロセッサの命令制御機構において、第 2 のフィールドに処理対象の他のレジスタを記載したレジスタ命令コードは、命令フィールドを特定の値とし、第 2 のフィールドのうちの処理対象の他のレジスタを記載した以外の部分に第 2 の命令フィールドに記載することにより表され、第 2 のフィールドに即値を記載した即値命令コードは、命令フィールドに特定の値以外の値が記載され、即値と処理対象の他のレジスタを使用する以外は同一の処理である少なくとも一部の即値命令コードと即値類似レジスタ命令コードには、命令フィールドと第 2 の命令フィールドに同一の値が割り当てられ、命令フィールドが特定の値であるかを

判定するレジスタ命令検出回路と、レジスタ命令検出回路の判定結果に従って、命令フィールドが特定の値である時には第2の命令フィールドが、命令フィールドが特定の値でない時には命令フィールドが、命令デコーダに入力されるように切り換える選択回路を備えることを特徴とする。

【0015】また、即値命令コード及び即値類似レジスタ命令コードと類似した処理であるが、即値及び処理対象の他のレジスタの値は使用しない第2のレジスタ命令コードの少なくとも一部には、命令フィールドに特定の値以外の第2の特定の値で、第2の命令フィールドに即値類似レジスタ命令コードの第2の命令フィールドの値と同一の値を割り当てるようにし、レジスタ命令検出回路が、命令フィールドが第2の特定の値であるかも判定するようにすることが望ましい。このような命令の例としては、レジスタ間転送命令がある。

【0016】本発明のプロセッサの命令制御機構においては、基本長命令を実行するモードと短縮長命令を実行するモードを切り換えて、両方の命令が実行できるようにする。そこで、短縮長命令コードの命令フィールドは、基本長命令コードの命令フィールドと同じ位置で、処理内容が類似の即値類似レジスタの第2の命令フィールドと同一の値とし、基本長命令コードを処理する基本長モードと短縮長命令コードを処理する短縮長モードを切り換えるモード切り換え回路を設け、命令デコーダは、モードに応じて解釈処理を変化させる。

【0017】ここで、短縮長モード時に、レジスタ命令検出回路が命令フィールドが特定の値であることを検出した時には、命令デコーダは、この命令コードを基本長命令コードとして処理するようにすれば、短縮命令を実行するモードにおいても、そのまま基本長命令が実行できるようにする。

【0018】

【発明の実施の形態】図6は、本発明の実施例における命令コードの構造を示す図であり、(1)が基本長命令を、(2)が短縮長命令を示し、それぞれ(a)がレジスタアドレッシング命令を、(b)が即値アドレッシング命令を示す。図示のように、本実施例の命令コードにおいては、短縮長命令及び基本長の即値アドレッシング命令は、図4の従来の命令コードと同じ構造であり、異なるのは基本長のレジスタアドレッシング命令の構造である。基本長のレジスタアドレッシング命令では、上位6ビットの第1の命令フィールドOP1の値が0000000又は000100であり、従来例においては使用されなかった下位11ビットに第2の命令フィールドOP2と第3の命令フィールドOP3が記載される。

【0019】図7は、実施例における基本長の即値アドレッシング命令のセットを示す図である。ここでは各命令の詳しい説明は省略するが、命令は46個あり、空白の部分は対応する命令が規定されておらず、斜線に部

分、すなわち000000と000100は基本長のレジスタアドレッシング命令であることを示す。図8は、基本長のレジスタアドレッシング命令のセットを示す図である。図示のように、基本長のレジスタアドレッシング命令の第1の命令フィールドOP1の値は000000又は000100であり、第2の命令フィールドOP2の値により処理の内容が規定される。ここで重要なのが第2の命令フィールドOP2の値は、類似の処理を行う基本長の即値アドレッシング命令の第1の命令フィールドOP1の値と同じである点である。例えば、第1の命令フィールドOP1の値が000000の基本長のレジスタアドレッシング命令の乗算命令MUL(Rs1, Rs2, Rd)の第2の命令フィールドOP2の値は110111であり、対応する基本長の即値アドレッシング命令MUL1(Rs1, #16, Rd)の第1の命令フィールドOP1の値と同じである。また、第1の命令フィールドOP1の値が000000の基本長のレジスタアドレッシング命令の加算命令ADD(Rs1, Rs2, Rd)の第2の命令フィールドOP2の値は001111であり、対応する基本長の即値アドレッシング命令ADDI(Rs1, #16, Rd)の第1の命令フィールドOP1の値と同じである。

【0020】ここで、第1の命令フィールドOP1の値が000100の基本長のレジスタアドレッシング命令についても、同様に、第2の命令フィールドOP2の値は、類似の処理を行う基本長の即値アドレッシング命令の第1の命令フィールドOP1の値と同じである。例えば、加算命令には、符号なし飽和处理付き32ビット加算(ADD)や、符号付き飽和处理付き32ビット加算(ADDS)や、オペランドを上位と下位で16ビットずつに分け、それぞれに符号付き飽和处理付きの16ビット加算を同時に行う加算(ADDHSS)や、オペランドを上位と下位で16ビットずつに分け、それぞれに符号なし飽和处理付きの16ビット加算を同時に行う加算

(ADDHUS)などの各種の加算処理がある。本実施例では、ADDSは、第1の命令フィールドOP1の値が000000の基本長のレジスタアドレッシング命令として規定し、ADDSとADDHSSとADDHUSは、第1の命令フィールドOP1の値が000001の基本長のレジスタアドレッシング命令として規定している。ADDSとADDHSSとADDHUSの第2の命令フィールドOP2の値は001111であり、第3の命令フィールドOP3の値を異ならせることにより、これらの処理を識別している。このように、同じような処理であるが、処理内容が細部で異なる処理を第3の命令フィールドOP3により規定している。

【0021】図7に示すように、第1の命令フィールドOP1は6ビットであり、そこでは最大64通りの命令を規定できる。従来は、この64通りで、即値アドレッシング命令を規定していた。レジスタアドレッシング命令については、OP1とOP2と合せて命令を規定して

10

20

30

40

50

いた。これに対して、本実施例では、46通りの即値アドレッシング命令が規定され、72通りのレジスタアドレッシング命令が規定されており、即値アドレッシング命令は62通りまで（2つの値をレジスタアドレッシング命令に割り当てるとした場合）、レジスタアドレッシング命令は第2の命令フィールドOP2だけで規定したとしても62通り（000000と000100除くとする）規定することが可能であり、第3の命令フィールドOP3も使用するとすれば更に32倍のレジスタアドレッシング命令を規定することが可能である。

【0022】図10は、短縮長命令の例を示す図であり、ここでは加算命令の例を示している。ここで重要なのは、短縮長命令の命令フィールドの値001111は、類似の処理を行う基本長の即値アドレッシング命令の第1の命令フィールドOP1の値及び基本長のレジスタアドレッシング命令の第2の命令フィールドOP2の値と同じである点である。また、上記のADD、ADDSS、ADDHSS及びADDHUSのいずれの加算命令であるかは、9ビット目と5ビット目の値の組合せで規定している。なお、短縮長命令においては、演算対象の第1のレジスタRs1と演算結果を格納する結果格納先レジスタRdは同じレジスタであり、Rs1/Rdで指示されるレジスタの値に対して処理を行い、その処理結果を同じレジスタに格納する。

【0023】図11と図12は、本発明の実施例のプロセッサの命令制御機構の構成を示す図である。基本長命令を処理するモードであるか、短縮長命令を処理するモードであるかは、プログラム中の命令により、モードレジスタに所定の値をセットすることにより設定されるようになっている。いずれのモードにおいても、メモリから32ビット分の命令コードが読み出され、命令保持用フリップフロップ（FF）41と42に一時的に保持される。命令保持用フリップフロップは、上位16ビットを保持するFF41と下位16ビットを保持するFF42で構成される。従って、基本長命令の場合には、第1の命令フィールドOP1と第1の演算対象レジスタRs1と格納先レジスタRdがFF41に保持され、第2の演算対象レジスタRs2と第2の命令フィールドOP2と第3の命令フィールドOP3又は即値#16がFF42に保持され、短縮長命令の場合には、一方の命令フィールドOPとRs/RdとRs2（又は即値#4）がFF41に保持され、他方のOPと第1のレジスタRs/Rdと第2のレジスタRs2（又は即値#4）がFF42に保持される。

【0024】まず、基本長命令を処理するモード時の動作について説明する。OP1はデコーダ43に入力され、その値が000100又は000000であるかが判定される。OP1が000100又は000000であればレジスタアドレッシング命令であり、000100と000000以外であれば即値アドレッシング命令である。デコーダ43は、この判定結果

とモード信号に従って命令モード信号を出力する。セクタ44には、OP1（FF41のOPと同じ）、OP2、FF42のOPが入力される。セクタ44は、命令モード信号に従って、レジスタアドレッシング命令であれば入力される信号のうちからOP2を選択し、即値アドレッシング命令であればOP1を選択して、命令デコーダ45に出力する。前述のように、レジスタアドレッシング命令のOP2と即値アドレッシング命令OP1は、類似の処理であれば同じ値であるから、命令デコーダ45は最大62通りの命令が識別できればよい。命令デコーダ45は入力されるOP1又はOP2を解釈して、演算器制御信号を出力する。

【0025】セクタ51には、FF41のRs1とRdとRs1/Rd及びFF42のRs1/Rdが入力され、セクタ52には、FF41のRs2（#4）及びFF42のRs2とRs2（#4）が入力され、セクタ53には、FF42の即値#16（Rs2とOP2とOP3）が入力される。セクタ51は、命令モード信号に従って、Rs1とRdをそれぞれレジスタ55のアドレスデコーダD（Rs1）とD（Rd）に入力する。ここで、フリップフロップFF54は、基本長命令を実行するモードの場合には通過状態になる。同様に、セクタ52は、命令モード信号に従って、Rs2をレジスタ55のアドレスデコーダD（Rs2）に入力する。更に、セクタ53は、命令モード信号に従って、FF42の即値#16を選択してFF56に出力する。なお、図では命令モード信号のセクタ51、52及び53への配線を省略してある。

【0026】命令モード信号が基本長命令を示す時には、カウンタ値発生回路68が4を発生し、それをプログラムカウンタ47に印加する。プログラムカウンタ47はFF48に保持された直前のプログラムカウンタ値にこの4を加えてプログラムカウンタの値を増加させる。また、セクタ49は、短縮長命令にのみ関係するセクタであり、基本長命令には関係しない。デコーダ50には、セクタ49の出力と命令モード信号の一部が入力される。デコーダ50は、基本長命令の場合には、命令モード信号の一部を選択し、それをRs2/即値選択信号として出力する。

【0027】図12に示すように、レジスタ55の読み書きポートP（Rs1）から出力された値は一方の内部バス62に出力される。Rs2/即値選択信号に従って、レジスタ55の読み書きポートP（Rs2）の出力とFF56の出力の一方が、他方の内部バス61に出力される。演算器63と64は、内部バス61と62上の値に対して、命令デコーダ45の出力する演算制御信号により指示された処理を行い、その結果を出力ポート65と66を介して更に別の内部バス67に出力する。内部バス67に出力された値は、適当なタイミングでレジスタ55の読み書きポートP（Rd）に書き込まれる。

【0028】以上のようにして、レジスタアドレッシング命令の場合には、Rs 1とRs 2で指示されたレジスタの値に対して演算が行われ、その結果がRdで指示されるレジスタに記憶され、即値アドレッシング命令の場合には、Rs 1で指示されたレジスタの値と即値# 16に対して演算が行われ、その結果がRdで指示されるレジスタに記憶される。

【0029】次に、短縮長命令を処理するモード時の動作について説明する。短縮長命令のFF 41に保持されるOPはOP 1と同じビットであり、デコーダ43に10 入力され、その値が000100又は000000であるかが判定される。OPが000100又は000000であれば短縮長命令を処理するモード時であっても、基本長のレジスタアドレッシング命令であると判定され、上記の基本長命令用の処理が行われる。ただし、短縮長命令に基本長命令を混在させる場合には、基本長命令は4バイトの境界、すなわち、FF 41のOP (OP 1) に保持されるように制限されているものとする。OPが000100又は000000でなければ短縮長命令であり、カウンタ値発生回路49は2を発生して、プログラムカウンタ47は2増加する。

【0030】デコーダ43は、判定結果とモード信号に従って命令モード信号を出力する。セクタ44は、短縮長命令の場合には、プログラムカウンタ47の出力をラッチしたFF 48の出力の最下位のビットを判定して、FF 41のOPを選択すべきか、FF 42のOPを選択すべきかを判定して出力する。具体的には、FF 48の出力の最下位のビットが「0」の時にはFF 41のOPを、「1」の時にはFF 42のOPを選択する。上記のように、基本長のレジスタアドレッシング命令と判定された場合には、FF 41のOP、すなわちOP 1を30 選択して出力する。命令デコーダ45はセクタ45の出力を解読して演算制御信号を発生させる。前述のように、短縮長命令のOPとレジスタアドレッシング命令のOP 2は、類似の処理であれば同じ値であるから、命令デコーダ45は基本長命令と同じ解読処理を行えばよい。

【0031】セクタ51は、短縮長命令で、FF 48の出力の最下位のビットが「0」の時には、FF 41のRs 1/Rdをレジスタ55のD (Rs 1) に、「1」の時には、FF 42のRs 1/Rdをレジスタ55のD 40 (Rs 1) に出力する。同時に、セクタ51の出力は、FF 54で一旦保持された後、レジスタ55のD (Rd) に入力される。これは、演算器65と66での処理後にその処理結果をRs 1/Rdで指示される読み書きポートP (Rd) に書き込むためである。なお、基本長命令の場合には、セクタ51は、FF 41のRs 1とRdをそれぞれレジスタ55のD (Rs 1) とD (Rd) に出力する。

【0032】同様に、セクタ52は、短縮長命令で、FF 48の出力の最下位のビットが「0」の時には、F 50

FF 41のRs 2 (#4) をレジスタ55のD (Rs 2) に、「1」の時には、FF 42のRs 2 (#4) をレジスタ55のD (Rs 2) に出力する。基本長命令の場合には、セクタ52は、FF 42のRs 2をレジスタ55のD (Rs 2) に出力する。

【0033】また、セクタ53は、短縮長命令で、FF 48の出力の最下位のビットが「0」の時には、FF 41のRs 2 (#4) を、「1」の時には、FF 42のRs 2 (#4) をレジスタ55のD (Rs 2) に出力する。前述のように、基本長命令の場合には、セクタ53は、FF 42の# 16を出力する。ただし、短縮長命令に基本長の即値アドレッシング命令を混在させることはできないので、短縮長命令を実行するモードで、セクタ53が# 16を出力することは有り得ない。FF 42は、# 16を保持して出力するので、短縮長命令の時には、Rs 2 (#4) の上位に12個の「0」を付加する。

【0034】図10に示したように、短縮長命令では、9ビット目と5ビット目(25ビット目と20ビット目)が共に「0」の時に即値アドレッシング命令である。セクタ49は、25ビット目、20ビット目、9ビット目及び5ビット目の値、すなわち、図においてFF 41とFF 42で斜線で示した部分のビットが入力され、プログラムカウンタの値に応じて、25ビット目と20ビット目、9ビット目と5ビット目のいずれかを選択し、2つの値が共に「0」の時にFF 56から出力される即値を選択し、「1」の時にレジスタ55の読み書きポートP (Rs 2) から出力されたRs 2で指示されるレジスタの値を選択するRs 2/即値選択信号を出力する。

【0035】後の処理は、基本長命令の場合と同じである。ここで、短縮長命令では、すべての基本長命令を規定することができないため、プログラムサイズを小さくするために短縮長命令を使用している場合でも、短縮長命令セットで規定されない命令を使用する場合には、基本長命令を使用する必要がある。また、短縮長命令では、レジスタを指定するオペランドRs 1、Rs 2及びRdのビット数が基本長命令より小さいため、基本長命令で使用できるレジスタをすべて使用することはできない。そのため、短縮長命令で使用できないレジスタを使用する場合には、基本長命令を使用する必要がある。本実施例では、上記の説明のように、短縮長命令を実行するモード時でも、第1の命令フィールドOP 1 (FF 41に保持されるOP) が000100又は000000である時には、自動的に基本長命令であると判定してそれに対応する処理が行われる。従って、短縮長命令を実行している途中で基本長命令(但し、レジスタアドレッシング命令のみ)を実行する場合にも、モードの切り換えを行う必要がなく、プログラムを簡単にできる。

【0036】

【発明の効果】以上のように、本発明によれば、命令コードの無駄を生ぜずに短縮長命令と基本長命令を混在させることができるため、プログラムサイズを効率よく圧縮することが可能になる。また、短縮長命令モード中に、同一命令について基本長命令と短縮長命令を混在させることができるため、すべてのレジスタを有効に使用することができる。

【0037】更に、本発明では、命令デコーダは共通の処理内容であるため、その構成を非常に簡単にでき、パイプラインの段数を増加させる必要がなく、命令数を増加させてもデコーダのハードウェアの増加を最小限にすることができる。

【図面の簡単な説明】

【図 1】レジスタ及び即値（イミディエート）アドレッシングモードの命令コードの従来の構成例を示す図である。

【図 2】プロセッサの命令制御機構の従来の構成例を示す図である。

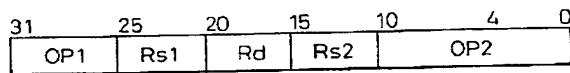
【図 3】短縮長命令を有する命令コードの従来の構成例（その 1）を示す図である。

【図 4】短縮長命令を有する命令コードの従来の構成例

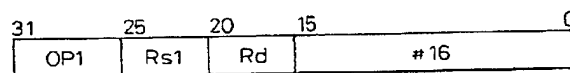
【図 1】

レジスタ及び即値（イミディエート）アドレッシングモードの命令コードの従来の構成例

(1) レジスタ命令

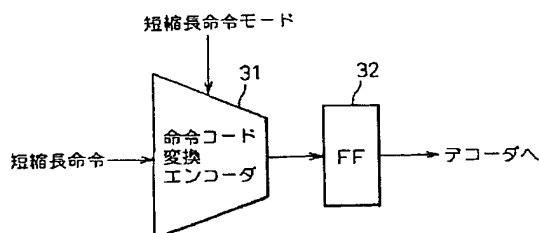


(2) 即値命令



【図 5】

従来例における変換機構



（その 2）を示す図である。

【図 5】短縮長命令を基本長命令に変換する従来の変換機構を示す図である。

【図 6】本発明の実施例の命令コードの構造を示す図である。

【図 7】実施例の基本長の即値アドレッシング命令の命令フィールド OP 1 のセットを示す図である。

【図 8】実施例の基本長のレジスタアドレッシング命令の命令フィールドのセットを示す図である。

【図 9】実施例の基本長のレジスタアドレッシング命令の命令フィールドのセットを示す図である。

【図 10】実施例の短縮長命令の例を示す図である。

【図 11】実施例の命令制御機構の構成を示す図である。

【図 12】実施例の命令制御機構の構成を示す図である。

【符号の説明】

4 1、4 2…命令保持用フリップフロップ（FF）

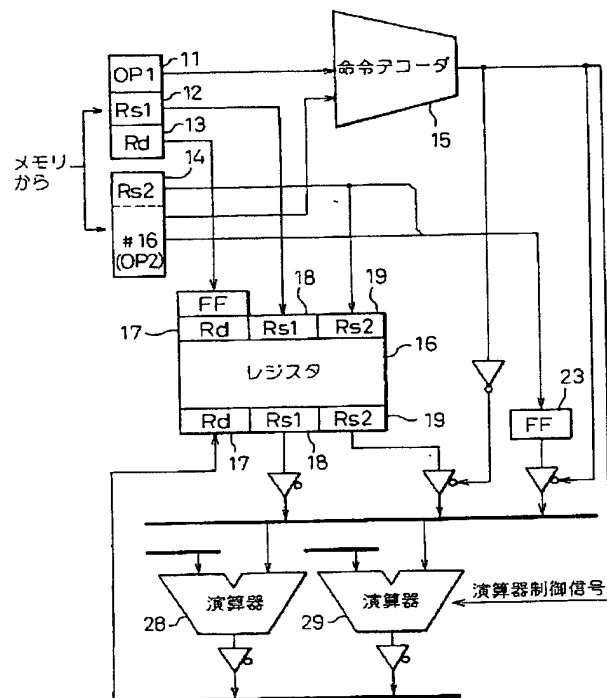
4 3…デコーダ

4 4…セクタ

4 5…命令デコーダ

【図 2】

プロセッサの命令制御機構の従来例

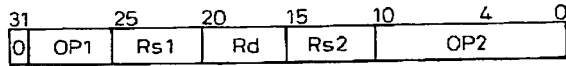


【図 3】

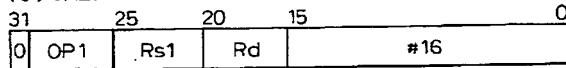
短縮長命令を有する命令コードの従来例 (その 1)

(1) 基本長命令

(a) レジスタアドレッシング

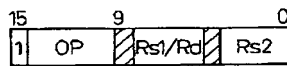


(b) 即値アドレッシング

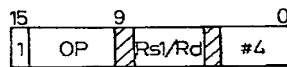


(2) 短縮長命令

(a) レジスタアドレッシング



(b) 即値アドレッシング

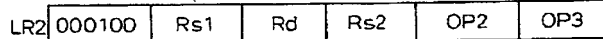
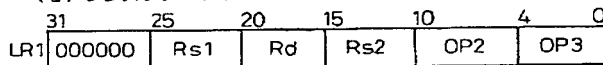


【図 6】

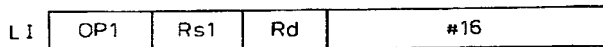
実施例の命令コードの構造

(1) 基本長命令

(a) レジスタアドレッシング

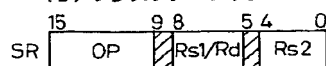


(b) 即値アドレッシング



(2) 短縮長命令

(a) レジスタアドレッシング



(b) 即値アドレッシング

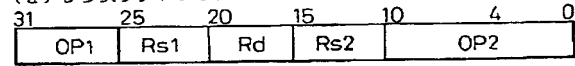


【図 4】

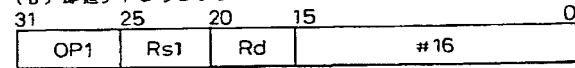
短縮長命令を有する命令コードの従来例 (その 2)

(1) 基本長命令

(a) レジスタアドレッシング

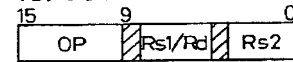


(b) 即値アドレッシング

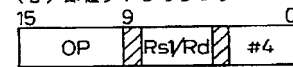


(2) 短縮長命令

(a) レジスタアドレッシング



(b) 即値アドレッシング



【図 8】

実施例のレジスタアドレッシング命令 (LR2)

操作	OP1	OP2	OP3
CMPL	Rs1, Rs2, Rd	000100	111111 000000
CMPL	Rs1, Rs2, Rd	000100	111110 000000
CMPL	Rs1, Rs2, Rd	000100	111101 000000
CMPL	Rs1, Rs2, Rd	000100	111100 000000
CMPL	Rs1, Rs2, Rd	000100	111011 000000
CMPL	Rs1, Rs2, Rd	000100	111010 000000
CMPL	Rs1, Rs2, Rd	000100	111001 000000
CMPL	Rs1, Rs2, Rd	000100	111000 000000
CMPL	Rs1, Rs2, Rd	000100	110101 000000
CMPL	Rs1, Rs2, Rd	000100	110100 000000
CMPL	Rs1, Rs2, Rd	000100	110011 000000
CMPL	Rs1, Rs2, Rd	000100	110010 000000
CMPL	Rs1, Rs2, Rd	000100	110001 000000
CMPL	Rs1, Rs2, Rd	000100	110000 000000
CMPL	Rs1, Rs2, Rd	000100	101111 000000
CMPL	Rs1, Rs2, Rd	000100	101110 000000
CMPL	Rs1, Rs2, Rd	000100	101101 000000
CMPL	Rs1, Rs2, Rd	000100	101100 000000
CMPL	Rs1, Rs2, Rd	000100	101011 000000
CMPL	Rs1, Rs2, Rd	000100	101010 000000
CMPL	Rs1, Rs2, Rd	000100	101001 000000
CMPL	Rs1, Rs2, Rd	000100	101000 000000
CMPL	Rs1, Rs2, Rd	000100	100111 000000
CMPL	Rs1, Rs2, Rd	000100	100110 000000
CMPL	Rs1, Rs2, Rd	000100	100101 000000
CMPL	Rs1, Rs2, Rd	000100	100100 000000
CMPL	Rs1, Rs2, Rd	000100	100011 000000
CMPL	Rs1, Rs2, Rd	000100	100010 000000
CMPL	Rs1, Rs2, Rd	000100	100001 000000
CMPL	Rs1, Rs2, Rd	000100	100000 000000
CMPL	Rs1, Rs2, Rd	000100	100111 000000
CMPL	Rs1, Rs2, Rd	000100	100110 000000
CMPL	Rs1, Rs2, Rd	000100	100101 000000
CMPL	Rs1, Rs2, Rd	000100	100100 000000
CMPL	Rs1, Rs2, Rd	000100	101011 000000
CMPL	Rs1, Rs2, Rd	000100	101010 000000
CMPL	Rs1, Rs2, Rd	000100	101001 000000
CMPL	Rs1, Rs2, Rd	000100	101000 000000
CMPL	Rs1, Rs2, Rd	000100	100111 000000
CMPL	Rs1, Rs2, Rd	000100	100110 000000
CMPL	Rs1, Rs2, Rd	000100	100101 000000
CMPL	Rs1, Rs2, Rd	000100	100100 000000
CMPL	Rs1, Rs2, Rd	000100	100011 000000
CMPL	Rs1, Rs2, Rd	000100	100010 000000
CMPL	Rs1, Rs2, Rd	000100	100001 000000
CMPL	Rs1, Rs2, Rd	000100	100000 000000
CMPL	Rs1, Rs2, Rd	000100	100111 000000
CMPL	Rs1, Rs2, Rd	000100	100110 000000
CMPL	Rs1, Rs2, Rd	000100	100101 000000
CMPL	Rs1, Rs2, Rd	000100	100100 000000
CMPL	Rs1, Rs2, Rd	000100	100011 000000
CMPL	Rs1, Rs2, Rd	000100	100010 000000
CMPL	Rs1, Rs2, Rd	000100	100001 000000
CMPL	Rs1, Rs2, Rd	000100	100000 000000
CMPL	Rs1, Rs2, Rd	000100	100111 000000
CMPL	Rs1, Rs2, Rd	000100	100110 000000
CMPL	Rs1, Rs2, Rd	000100	100101 000000
CMPL	Rs1, Rs2, Rd	000100	100100 000000
CMPL	Rs1, Rs2, Rd	000100	100011 000000
CMPL	Rs1, Rs2, Rd	000100	100010 000000
CMPL	Rs1, Rs2, Rd	000100	100001 000000
CMPL	Rs1, Rs2, Rd	000100	100000 000000
CMPL	Rs1, Rs2, Rd	000100	100111 000000
CMPL	Rs1, Rs2, Rd	000100	100110 000000
CMPL	Rs1, Rs2, Rd	000100	100101 000000
CMPL	Rs1, Rs2, Rd	000100	100100 000000
CMPL	Rs1, Rs2, Rd	000100	100011 000000
CMPL	Rs1, Rs2, Rd	000100	100010 000000
CMPL	Rs1, Rs2, Rd	000100	100001 000000
CMPL	Rs1, Rs2, Rd	000100	100000 000000
CMPL	Rs1, Rs2, Rd	000100	100111 000000
CMPL	Rs1, Rs2, Rd	000100	100110 000000
CMPL	Rs1, Rs2, Rd	000100	100101 000000
CMPL	Rs1, Rs2, Rd	000100	100100 000000
CMPL	Rs1, Rs2, Rd	000100	100011 000000
CMPL	Rs1, Rs2, Rd	000100	100010 000000
CMPL	Rs1, Rs2, Rd	000100	100001 000000
CMPL	Rs1, Rs2, Rd	000100	100000 000000

【図 7】

実施例の基本長命令のOP 1

上位	111	110	101	100	011	010	001	000
下位	111 CMPLI Rs1, #16,Rd	MULI Rs1, #16,Rd	DIVI Rs1, #16,Rd	CUTI #16,Rd	LDSD @(Rs1,#16), Rd		ADDI Rs1, #16,Rd	
110	CMPLI Rs1, #16,Rd	MULI Rs1, #16,Rd	DIVI Rs1, #16,Rd	CUTSI #16,Rd	LDHD @(Rs1,#16), Rd		SUBI Rs1, #16,Rd	
101	CMPLTI Rs1, #16,Rd	CMPEQI Rs1, #16,Rd	MODI Rs1, #16,Rd	LDI #16,Rd	LDWD @(Rs1,#16), Rd		LDIHI Rs1, #16,Rd	
100	CMPLTI Rs1, #16,Rd	CMPEQI Rs1, #16,Rd	MODI Rs1, #16,Rd		LDHWD @(Rs1,#16), Rd		LDDI @(Rs1,#16), Rd	
011	CMPEI Rs1, #16,Rd	BRT Rs1, #16	SRLI Rs1, #16,Rd	SRLHI Rs1, #16,Rd	STBD Rd @(Rs1,#16)		ANDI Rs1, #16,Rd	
010	CMPEI Rs1, #16,Rd	BRT Rs1, #16	SRAI Rs1, #16,Rd	SRAHI Rs1, #16,Rd	STHD Rd @(Rs1,#16)		ORI Rs1, #16,Rd	
001	CMPTI Rs1, #16,Rd	BR #16	SLLI Rs1, #16,Rd	SLLHI Rs1, #16,Rd	STWD Rd @(Rs1,#16)		XORI Rs1, #16,Rd	
000	CMPTI Rs1, #16,Rd	CALL #16,Rd	ROLI Rs1, #16,Rd	RORI Rs1, #16,Rd	STHWD Rd @(Rs1,#16)			

【図 9】

【図 10】

実施例のレジスタアドレッシング命令 (L R 1)

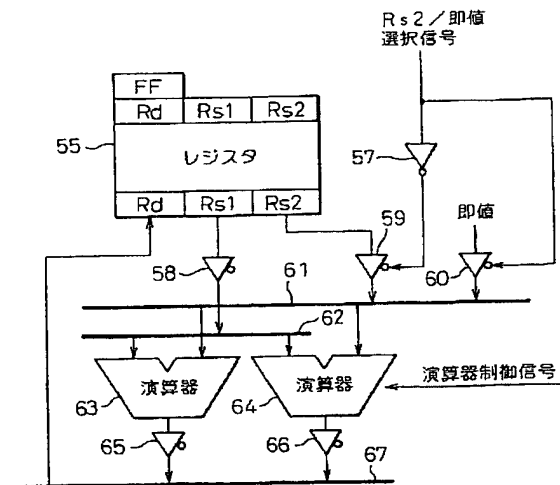
CALLR	@Rs1, Rd	000000	110000	000001
JMP	@Rs1,	000000	110000	100001
MAC	Rs1, Rs2	000000	110111	000000
MACH	Rs1, Rs2	000000	110111	100000
MULH	Rs1, Rs2	000000	110111	110000
FMACSS	Rs1, Rs2, Rd	000000	110111	101000
FMULSS	Rs1, Rs2, Rd	000000	110111	110000
HALT		000000	011001	000011
HALTB		000000	011001	100011
ADDSS	Rs1, Rs2, Rd	000000	001111	001000
ADDHSS	Rs1, Rs2, Rd	000000	001111	101000
ADDHUS	Rs1, Rs2, Rd	000000	001111	111000
AVEH	Rs1, Rs2, Rd	000000	001111	100000
SUBSS	Rs1, Rs2, Rd	000000	001110	001000
SUBHSS	Rs1, Rs2, Rd	000000	001110	101000
SUBHUS	Rs1, Rs2, Rd	000000	001110	111000
NEG	Rs1, Rd	000000	001110	000010
NOT	Rs1, Rd	000000	001001	000001
NOP		000000	000000	000011
CLRACC		000000	000000	100011
EXTZB	Rs1, Rd	000000	001011	000001
EXTZH	Rs1, Rd	000000	001011	100001
SATS	Rs1, Rs2, Rd	000000	001101	001000
SATU	Rs1, Rs2, Rd	000000	001101	011000
SATHS	Rs1, Rs2, Rd	000000	001101	101000
SATHU	Rs1, Rs2, Rd	000000	001101	111000
MOV	Rs1, Rd	000000	001101	000001

実施例の短縮長命令の例

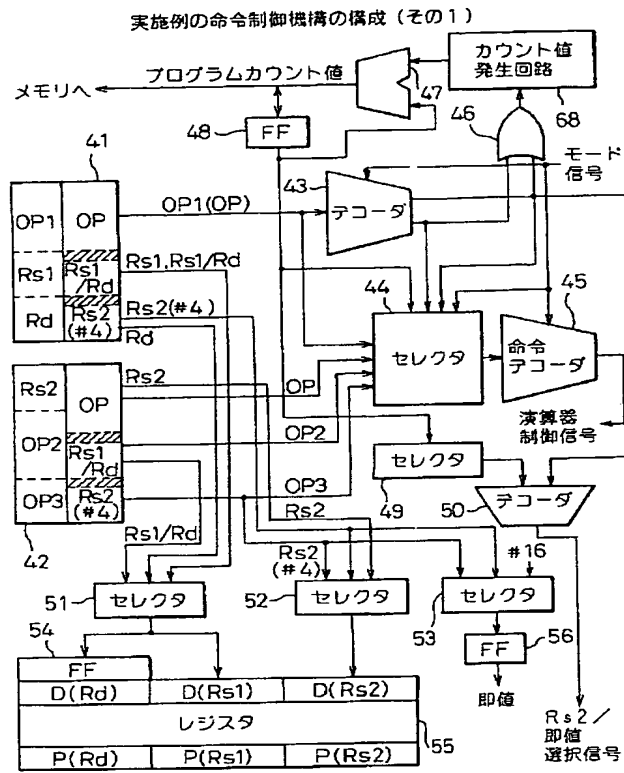
操作	フォーマット	コード
ADD R1, #4	SI (001111	0**** 0****)
ADD R1, R2	SR (001111	0**** 0****)
ADDSS R1, R2	SR (001111	0**** 1****)
ADDHSS R1, R2	SR (001111	1**** 0****)
ADDHUS R1, R2	SR (001111	1**** 1****)

【図 12】

実施例の命令制御機構の構成 (その2)



【図 1 1】



THIS PAGE BLANK (USPT